Introduction of AI (CS103)- 09 Machine Learning Algorithms 2

Jimmy Liu 刘江

2023-11-17

Advantages and Disadvantages of SVM

Advantages:

1. Effective in high-dimensional spaces.
2. Memory efficient.
3. Versatile: Different Kernel functions can be specified.

Disadvantages:

1. Not suitable for large datasets due to high training time.
2. Less effective on noisier datasets with overlapping classes.

Example Comparison

Table: Distance from each sample in the myopia dataset to the hyperplane

Method	w	b	x	d
SVM	(–0.0150, –0.1445, 0.0780)	–2.6062	(3.7, –10.5, 27.49)	6.07
			(4.8, 0.13, 21.75)	6.07
			(4.9, 1.25, 22.31)	6.79
			(3.8, –6.38, 24.71)	1.13
			(5.0, –0.38, 22.79)	5.14
Perception	(–8.4, –35.38, –6.21)	0	(3.7, –10.5, 27.49)	4.57
			(4.8, 0.13, 21.75)	4.91
			(4.9, 1.25, 22.31)	6.10
			(3.8, –6.38, 24.71)	1.07
			(5.0, –0.38, 22.79)	4.64

What is a support vector machine? What is the core?

Support vector machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks.

The core idea is to find an optimal hyperplane in the feature space to maximize the spacing of different classes, while ensuring that the separated hyperplane can correctly classify the training data.

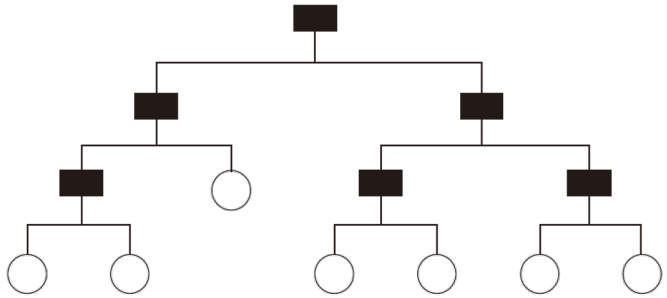
This optimal hyperplane is called the maximum interval hyperplane.

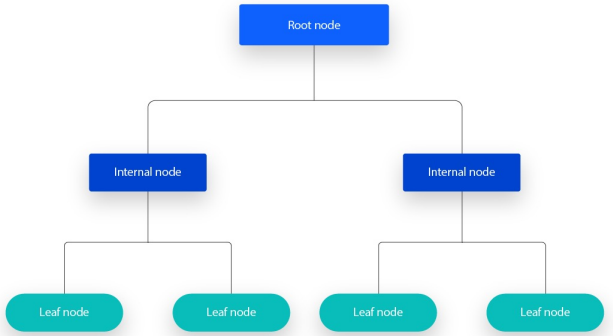
Lecture 9

1. Reviews of Lecture 8
2. Decision Tree
3. Ensemble Learning
4. Unsupervised Learning

Decision Tree

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.



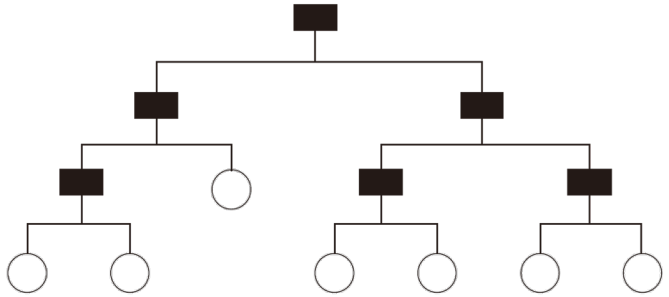


Decision Tree

The principle for decision tree is **divide-and-conquer**.

Divide and conquer is an algorithm design paradigm. A divide-and-conquer algorithm recursively breaks down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly.

The solutions to the sub-problems are then combined to give a solution to the original problem.



Decision Tree Algorithm

Pseudo code

Part 1

Input: Training Set D = $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$;
Feature Set A = $\{a_1, a_2, \dots, a_M\}$

Process:

Function TreeGenerate(D, A)

1. Generate a Node
2. **If** the samples in D belongs to the same category C **then**
3. Note this Node as C class Node **return**
4. **end if**
5. **If** A is null **or** the samples in D has same values for A **then**
6. Mark this Node as the leaf node with the category has most samples, **return**
7. **end if**
8. Select the best separation a_i from A according to the feature selection method chosen

Decision Tree Algorithm

Pseudo code

Part 2

9. **For** each value a_t^t in a_t **do**
10. create a branch for the Node, let D_t being the subset for D when $a_t^t = a_t$
11. **if** D_t is null **then**
12. set branch node as the category with most samples in D
13. **else**
14. set TreeGenerate($D_i, A \backslash \{a_t\}$) as the branch node
15. **end if**
16. **end for**

Output: A Decision Tree

Feature Selection for Decision Tree

To partition the node with more samples in the nodes shared the same features. Some feature selection method are used: information entropy, gain ratio or Gini index.

（1）Information entropy (ID3)

Information entropy is a method for measuring the purity. Assume in Aggregation D, the k_{th} sample ratio is defined as p_k (k = 1, 2, …N) , the information entropy should be ：

$$\mathrm{Ent}(D) = -\sum_{k=1}^{N} p_k \log_2 p_k$$

$$\mathrm{Gain}\ (D, a) = \mathrm{Ent}(D) - \sum_{i=1}^{M} \frac{|\ D^i\ |}{|\ D\ |} \mathrm{Ent}(D^i)$$

Feature selection for Decision Tree

（2）Gain ratio (C4.5)

Information entropy may have bias on features with more samples, to minimize this effect to avoid over sampling, we can use gain ratio to divide the features

$$\text{Gainratio}(D,a) = \frac{\text{Gain}(D,a)}{\text{IV}(a)}$$

$$\text{IV}(a) = -\sum_{i=1}^{M} \frac{|D^i|}{|D|} \log_2 \frac{|D^i|}{|D|}$$

Feature selection for Decision Tree

（3）Gini index

CART [Breiman et al., 1984] have used Gini index to do the division:

$$\text{Gini}_\text{i}\text{ndex}\,(D,a) = \sum_{i=1}^{M} \frac{|D^i|}{|D|}\text{Gini}\left(D^i\right) = 1 - \sum_{k=1}^{|N|} p_k^2$$

Decision Tree Example

Here is the training sample with 24 patients to judge whether they should take contact lens. The features includes age, vision situation, astigmatism, changes of the tear amounts.

Please use information entropy to do the decision tree.

No	Age	Eyesight	Astigmatism	Change of Tears	Contact lens Category
1	Young	myopia	F	Less	Unsuitable
2	Young	myopia	F	Normal	Soft
3	Young	myopia	T	Less	Unsuitable
4	Young	myopia	T	Normal	Hard
5	Young	hyperopia	F	Less	Unsuitable
6	Young	hyperopia	F	Normal	Soft
7	Young	hyperopia	T	Less	Unsuitable
8	Young	hyperopia	T	Normal	Hard
9	Before presbyopia	myopia	F	Less	Unsuitable
10	Before presbyopia	myopia	F	Normal	Soft
11	Before presbyopia	myopia	T	Less	Unsuitable
12	Before presbyopia	myopia	T	Normal	Hard
13	Before presbyopia	hyperopia	F	Less	Unsuitable
14	Before presbyopia	hyperopia	F	Normal	Soft
15	Before presbyopia	hyperopia	T	Less	Unsuitable
16	Before presbyopia	hyperopia	T	Normal	Unsuitable
17	presbyopia	myopia	F	Less	Unsuitable
18	presbyopia	myopia	F	Normal	Unsuitable
19	presbyopia	myopia	T	less	Unsuitable
20	presbyopia	myopia	T	Normal	Hard
21	presbyopia	hyperopia	F	Less	Unsuitable
22	presbyopia	hyperopia	F	Normal	Soft
23	presbyopia	hyperopia	T	Less	Unsuitable
24	presbyopia	hyperopia	T	Normal	Unsuitable

Patient sample

Decision Tree

No	Age	Eyesight	Astigmatism	Change of Tears	Contact lens Category
1	Young	myopia	F	Less	Unsuitable
2	Young	myopia	F	Normal	Soft
3	Young	myopia	T	Less	Unsuitable
4	Young	myopia	T	Normal	Hard
5	Young	hyperopia	F	Less	Unsuitable
6	Young	hyperopia	F	Normal	Soft
7	Young	hyperopia	T	Less	Unsuitable
8	Young	hyperopia	T	Normal	Hard
9	Before presbyopia	myopia	F	Less	Unsuitable
10	Before presbyopia	myopia	F	Normal	Soft
11	Before presbyopia	myopia	T	Less	Unsuitable
12	Before presbyopia	myopia	T	Normal	Hard
13	Before presbyopia	hyperopia	F	Less	Unsuitable
14	Before presbyopia	hyperopia	F	Normal	Soft
15	Before presbyopia	hyperopia	T	Less	Unsuitable
16	Before presbyopia	hyperopia	T	Normal	Unsuitable
17	presbyopia	myopia	F	Less	Unsuitable
18	presbyopia	myopia	F	Normal	Unsuitable
19	presbyopia	myopia	T	less	Unsuitable
20	presbyopia	myopia	T	Normal	Hard
21	presbyopia	hyperopia	F	Less	Unsuitable
22	presbyopia	hyperopia	F	Normal	Soft
23	presbyopia	hyperopia	T	Less	Unsuitable
24	presbyopia	hyperopia	T	Normal	Unsuitable

We should start from the root node，rigid has 4/24，softness takes 5/24，unsuitable makes 15/24, we can do the calculation:

$$\mathrm{Ent}(D) = -\sum_{k=1}^{3} p_k \log_2 p_k$$

$$= -\left(\frac{5}{24}\log_2\frac{5}{24} + \frac{4}{24}\log_2\frac{4}{24} + \frac{15}{24}\log_2\frac{15}{24}\right)$$

$$= 1.326$$

Decision Tree

No	Age	Eyesight	Astigmatism	Change of Tears	Contact lens Category
1	Young	myopia	F	Less	Unsuitable
2	Young	myopia	F	Normal	Soft
3	Young	myopia	T	Less	Unsuitable
4	Young	myopia	T	Normal	Hard
5	Young	hyperopia	F	Less	Unsuitable
6	Young	hyperopia	F	Normal	Soft
7	Young	hyperopia	T	Less	Unsuitable
8	Young	hyperopia	T	Normal	Hard
9	Before presbyopia	myopia	F	Less	Unsuitable
10	Before presbyopia	myopia	F	Normal	Soft
11	Before presbyopia	myopia	T	Less	Unsuitable
12	Before presbyopia	myopia	T	Normal	Hard
13	Before presbyopia	hyperopia	F	Less	Unsuitable
14	Before presbyopia	hyperopia	F	Normal	Soft
15	Before presbyopia	hyperopia	T	Less	Unsuitable
16	Before presbyopia	hyperopia	T	Normal	Unsuitable
17	presbyopia	myopia	F	Less	Unsuitable
18	presbyopia	myopia	F	Normal	Unsuitable
19	presbyopia	myopia	T	less	Unsuitable
20	presbyopia	myopia	T	Normal	Hard
21	presbyopia	hyperopia	F	Less	Unsuitable
22	presbyopia	hyperopia	F	Normal	Soft
23	presbyopia	hyperopia	T	Less	Unsuitable
24	presbyopia	hyperopia	T	Normal	Unsuitable

We then calculation the information entropy for each features. Take “age” as example，it has 3 different values that can be noted as D_1(age=young);D_2(age=before presbyopia); D_3 (age= presbyopia).

$$\mathrm{Ent}(D_1) = -\left(\frac{2}{8}\log_2\frac{2}{8}+\frac{2}{8}\log_2\frac{2}{8}+\frac{4}{8}\log_2\frac{4}{8}\right) = 1.5$$

$$\mathrm{Ent}(D_2) = -\left(\frac{2}{8}\log_2\frac{2}{8}+\frac{1}{8}\log_2\frac{1}{8}+\frac{5}{8}\log_2\frac{5}{8}\right) = 1.2987$$

$$\mathrm{Ent}(D_3) = -\left(\frac{1}{8}\log_2\frac{1}{8}+\frac{1}{8}\log_2\frac{1}{8}+\frac{6}{8}\log_2\frac{6}{8}\right) = 1.0612$$

Decision Tree

No	Age	Eyesight	Astigmatism	Change of Tears	Contact lens Category
1	Young	myopia	F	Less	Unsuitable
2	Young	myopia	F	Normal	Soft
3	Young	myopia	T	Less	Unsuitable
4	Young	myopia	T	Normal	Hard
5	Young	hyperopia	F	Less	Unsuitable
6	Young	hyperopia	F	Normal	Soft
7	Young	hyperopia	T	Less	Unsuitable
8	Young	hyperopia	T	Normal	Hard
9	Before presbyopia	myopia	F	Less	Unsuitable
10	Before presbyopia	myopia	F	Normal	Soft
11	Before presbyopia	myopia	T	Less	Unsuitable
12	Before presbyopia	myopia	T	Normal	Hard
13	Before presbyopia	hyperopia	F	Less	Unsuitable
14	Before presbyopia	hyperopia	F	Normal	Soft
15	Before presbyopia	hyperopia	T	Less	Unsuitable
16	Before presbyopia	hyperopia	T	Normal	Unsuitable
17	presbyopia	myopia	F	Less	Unsuitable
18	presbyopia	myopia	F	Normal	Unsuitable
19	presbyopia	myopia	T	less	Unsuitable
20	presbyopia	myopia	T	Normal	Hard
21	presbyopia	hyperopia	F	Less	Unsuitable
22	presbyopia	hyperopia	F	Normal	Soft
23	presbyopia	hyperopia	T	Less	Unsuitable
24	presbyopia	hyperopia	T	Normal	Unsuitable

In that way, the information entropy for age is:

$$\text{Gain}(D, \text{Age}) = \text{Ent}(D) - \sum_{i=1}^{3} \frac{|D^i|}{|D|} \text{Ent}(D^i)$$

$$= 1.326 - \left(\frac{8}{24} \times 1.5 + \frac{8}{24} \times 1.2987 + \frac{8}{24} \times 1.0612\right)$$

$$= 0.0393$$

Similarly, we can calculate other information entropy for different features.

$$\text{Gain}(D, \text{Eyesight}) = 0.0395$$

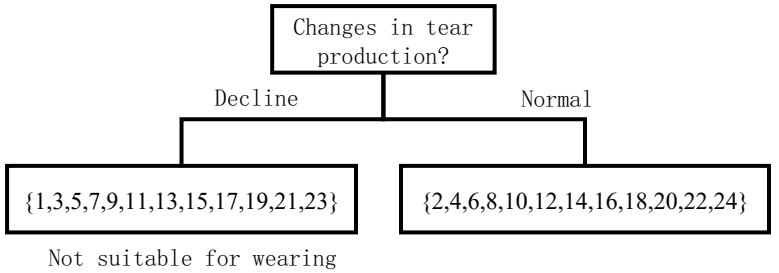
$$\text{Gain}(D, \text{Astigmatism}) = 0.3770$$

$$\text{Gain}(D, \text{Change of Tears}) = 0..5487$$

Decision Tree

No	Age	Eyesight	Astigmatism	Change of Tears	Contact lens Category
1	Young	myopia	F	Less	Unsuitable
2	Young	myopia	F	Normal	Soft
3	Young	myopia	T	Less	Unsuitable
4	Young	myopia	T	Normal	Hard
5	Young	hyperopia	F	Less	Unsuitable
6	Young	hyperopia	F	Normal	Soft
7	Young	hyperopia	T	Less	Unsuitable
8	Young	hyperopia	T	Normal	Hard
9	Before presbyopia	myopia	F	Less	Unsuitable
10	Before presbyopia	myopia	F	Normal	Soft
11	Before presbyopia	myopia	T	Less	Unsuitable
12	Before presbyopia	myopia	T	Normal	Hard
13	Before presbyopia	hyperopia	F	Less	Unsuitable
14	Before presbyopia	hyperopia	F	Normal	Soft
15	Before presbyopia	hyperopia	T	Less	Unsuitable
16	Before presbyopia	hyperopia	T	Normal	Unsuitable
17	presbyopia	myopia	F	Less	Unsuitable
18	presbyopia	myopia	F	Normal	Unsuitable
19	presbyopia	myopia	T	less	Unsuitable
20	presbyopia	myopia	T	Normal	Hard
21	presbyopia	hyperopia	F	Less	Unsuitable
22	presbyopia	hyperopia	F	Normal	Soft
23	presbyopia	hyperopia	T	Less	Unsuitable
24	presbyopia	hyperopia	T	Normal	Unsuitable

Get two different node，
do further division.



Decision Tree

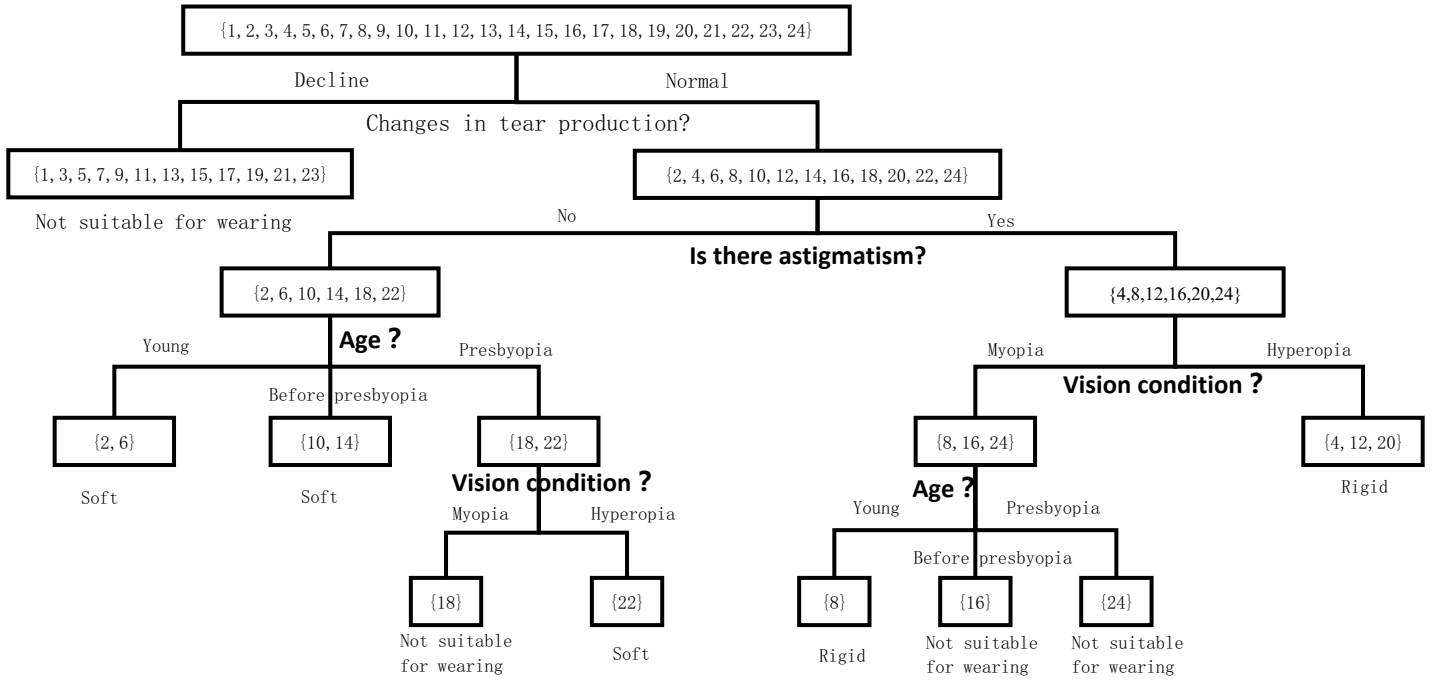
No	Age	Eyesight	Astigmatism	Change of Tears	Contact lens Category
1	Young	myopia	F	Less	Unsuitable
2	Young	myopia	F	Normal	Soft
3	Young	myopia	T	Less	Unsuitable
4	Young	myopia	T	Normal	Hard
5	Young	hyperopia	F	Less	Unsuitable
6	Young	hyperopia	F	Normal	Soft
7	Young	hyperopia	T	Less	Unsuitable
8	Young	hyperopia	T	Normal	Hard
9	Before presbyopia	myopia	F	Less	Unsuitable
10	Before presbyopia	myopia	F	Normal	Soft
11	Before presbyopia	myopia	T	Less	Unsuitable
12	Before presbyopia	myopia	T	Normal	Hard
13	Before presbyopia	hyperopia	F	Less	Unsuitable
14	Before presbyopia	hyperopia	F	Normal	Soft
15	Before presbyopia	hyperopia	T	Less	Unsuitable
16	Before presbyopia	hyperopia	T	Normal	Unsuitable
17	presbyopia	myopia	F	Less	Unsuitable
18	presbyopia	myopia	F	Normal	Unsuitable
19	presbyopia	myopia	T	less	Unsuitable
20	presbyopia	myopia	T	Normal	Hard
21	presbyopia	hyperopia	F	Less	Unsuitable
22	presbyopia	hyperopia	F	Normal	Soft
23	presbyopia	hyperopia	T	Less	Unsuitable
24	presbyopia	hyperopia	T	Normal	Unsuitable

For less tears, none of the samples was suitable to wear, with no need to do division. On the other hand, the normal change for the tears have {2,4,6,8,10,12,14,16,18,20,22,24} these 12 samples have different categories for the presbyopia. We can calculate the information entropy one by one for the features.

$$\text{Gain}(D_1, \text{Age}) = 0.2212$$
$$\text{Gain}(D_1, \text{Eyesight}) = 0.0954$$
$$\text{Gain}(D_1, \text{Astigmatism}) = 0.7704$$

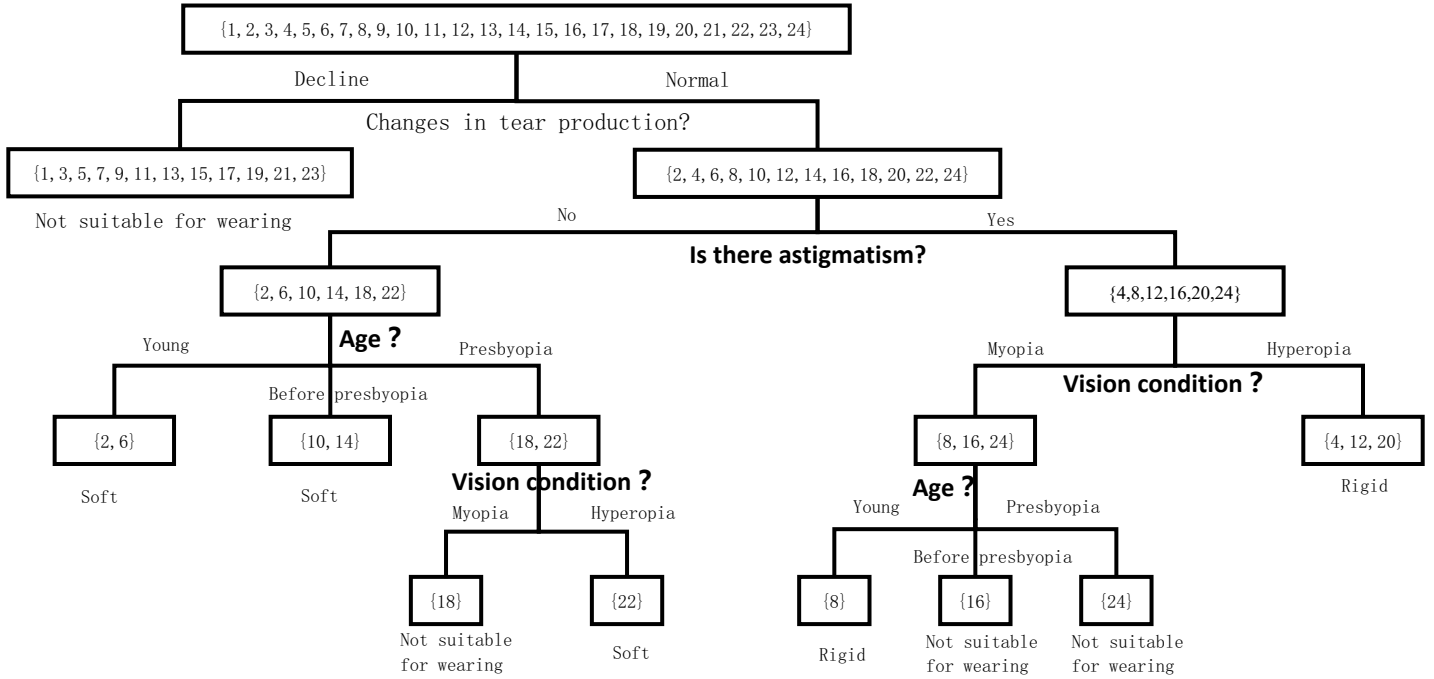
Decision Tree

Here, whether patients have astigmatism achieve the greatest information entropy, we select this branch to go further and get the final decision tree as below:



Decision Tree

We can see that decision tree is comprehensible and explainable. We can choose the gain ratio and Gini index according to actual situations.



ID3 Decision Trees May Overfit

Overfit Scenarios:

Overly complex tree structure: Decision trees tend to produce very deep trees during training, with a small number of samples per leaf node. Such tree structures can fit training data perfectly, but may perform poorly on unseen data. Overly split trees tend to remember noise and randomness in training data, rather than capturing the true pattern in the data. Too many features: If the data set contains a large number of features, the decision tree is prone to choose multiple features to split, resulting in a complex tree. Complex trees are easier to overfit because they can more easily adapt to noise in the training data.

Solution:

Tree complexity can be controlled through techniques such as pruning, or integrated methods such as random forests can be used to improve performance and generalization.

Lecture 9

1. Reviews of Lecture 8
2. Decision Tree
3. Ensemble Learning
4. Unsupervised Learning

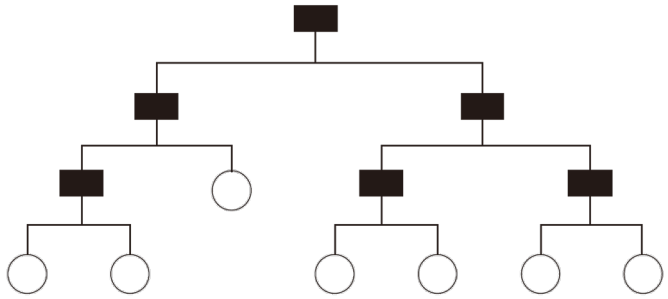
Overview of Ensemble Learning

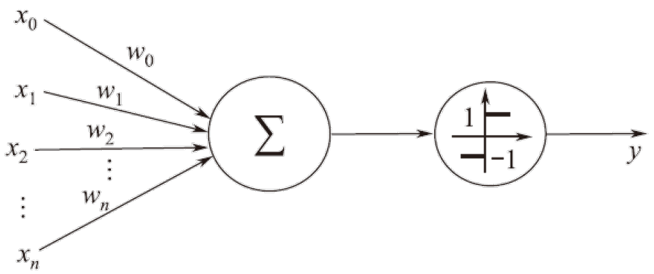
- Ensemble Learning involves the combination of multiple models to improve the overall performance. It aims to produce better predictive results compared to a single model.

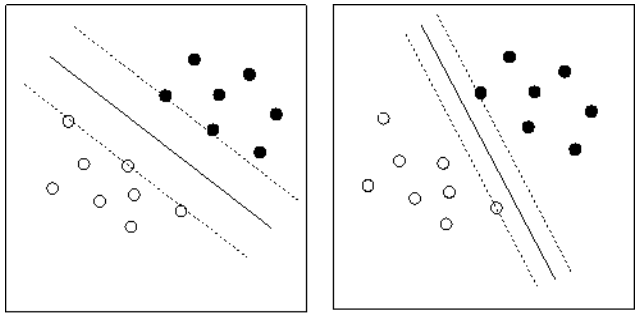
- The primary goal is to reduce overfitting, improve accuracy, and enhance the robustness of the model. It offers a way to mitigate the limitations of individual algorithms.

Base Models

- Base models are individual learning algorithms that are combined in ensemble methods. Commonly used base models include Decision Trees, SVMs, and Neural Networks.







Ensemble Methods

- These are strategies for combining base models. Major ensemble methods are Bagging, Boosting, and Stacking.
- For example, Bootstrap Aggregating (Bagging) involves creating multiple subsets of the original dataset and training a model on each. The final output is averaged (for regression) or voted (for classification).
- Random Forest is the classic algorithm for the Bagging method.

Bagging

- The Bagging algorithm can be divided into the following three steps.
  - Step 1: Given a data set containing K samples, K samples can be randomly put back and taken out to form a sampling set (some samples may not appear, some samples may appear multiple times), repeated K times, and K training sets containing K samples are generated.
  - Step 2: Each sample set corresponds to a training data set, and the corresponding individual learner is trained.
  - Step 3: Average the results of K individual learners according to the same weight (voting strategy for classification, mean for regression).

Random Forest

Random Forest builds multiple decision trees during training and merges them to produce more accurate and stable predictions. Each tree is built on a subset of the data, making the ensemble robust to outliers and noise.

Randomness are exhibited in two ways:

Bootstrap Sampling: Each sample set is randomly chosen with replacement.

Random Feature Selection: A random subset of features is used for each decision tree.

Random Forest

Detailed process of random forest

① The Bagging algorithm is used to construct T sample sets

② Train T decision trees in parallel. Each sample has M attributes, when node of the decision tree needs to be split, m attributes are randomly selected from the M attributes, and the conditions are satisfied m << M.

③ T decision trees, corresponding to T results (classification is the category result, regression is the numerical result), the final result by voting/averaging to give the final random forest result.

Random Forest

Randomness in a random forest

1: Using the self-sampling method, the samples of each sampling set are put back to random selection;
2: The attribute randomness of the feature subset is selected by random.

Pros and Cons

- **Pros**
  - 1. Improved Accuracy
  - 2. Robustness to Overfitting from particular feature
  - 3. Better Generalization
- **Cons**
  - 1. Increased Computational Cost
  - 2. Complexity in Interpretation
    - Overfitting: It has been proven to overfit in certain noisy classification or regression problems.(Random Forest)
    - Attribute Bias: For data with attributes of different scales or categories, attributes with more divisions have a greater impact on Random Forest. As a result, the attribute weights produced on such data are unreliable.

Summary： What is Ensemble learning? What are its advantages?

Ensemble learning is a machine learning technique that combines the predictions of **multiple basic models** (also known as weak models or base classifiers) to build a more powerful model to improve the model's performance and generalization ability.

The benefits of integrated learning include the following:

Improved generalization

Reducing contingency

Dealing with complex problems:

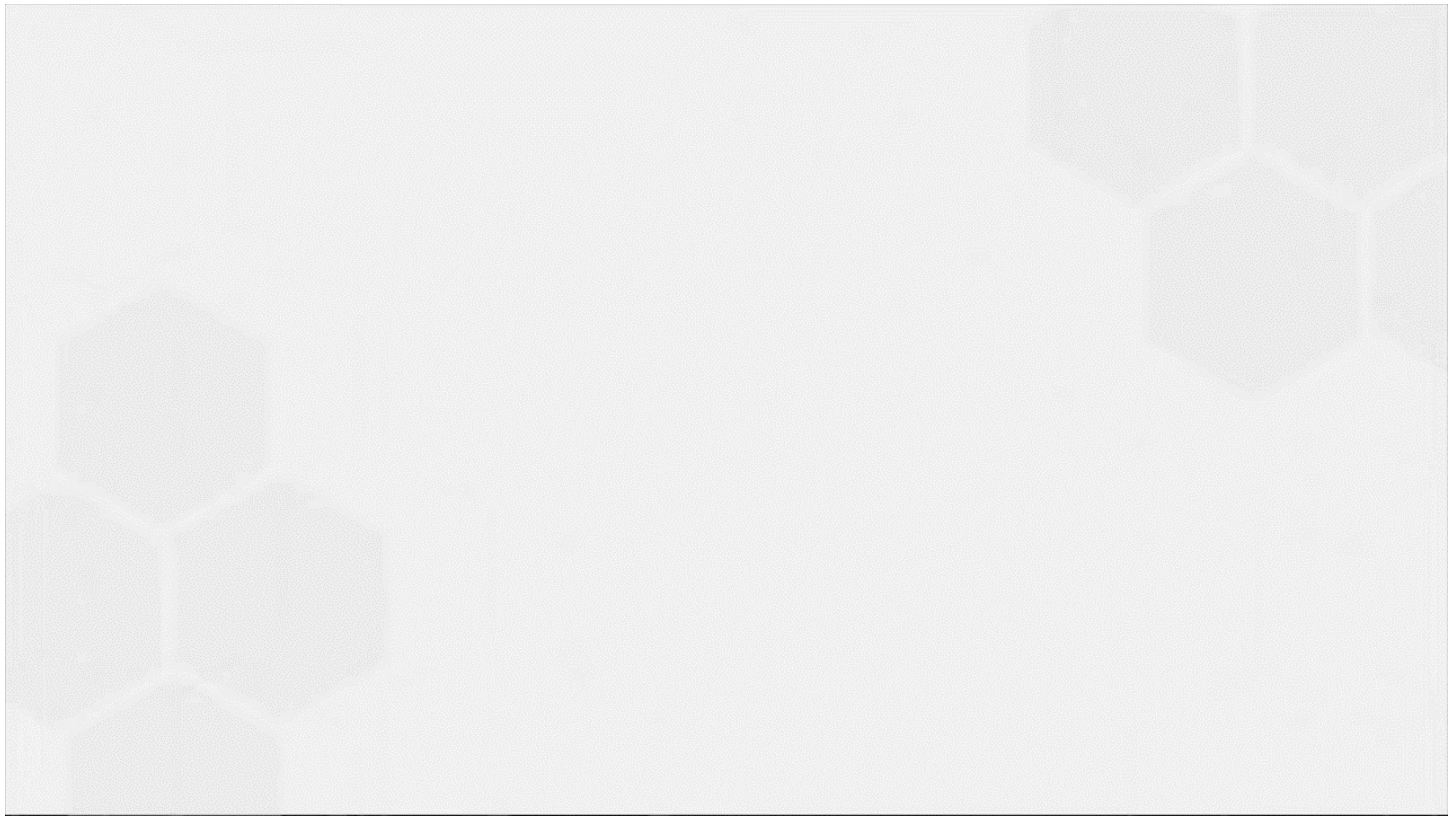
Adapt to different models

Lecture 9

1. Reviews of Lecture 8
2. Decision Tree
3. Ensemble Learning
4. Unsupervised Learning

Unsupervised Learning

Supervised vs. Unsupervised Machine Learning: What's the Difference ?

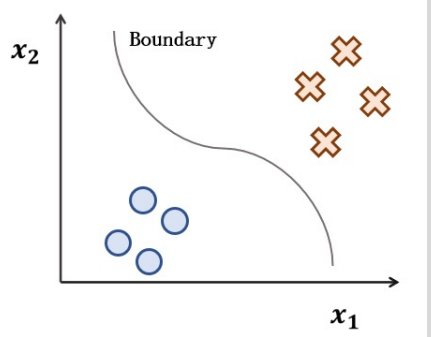


video source: https://www.youtube.com/watch?v=rHeaoaiBM6Y

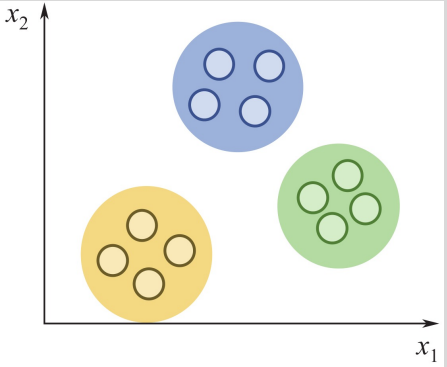
Unsupervised Learning

Unsupervised Learning

- In previous lectures, we have seen some supervised learning methods. Given some labelled data (discrete or continuous), we aim at training some reliable models to predict the labels of new data that have never been seen during training. However, this is not always the case in real life.
- If there is **no label**, can we still learn something from data? Today, we are going to see some unsupervised learning methods. The most popular one is Clustering.
- The goal of unsupervised learning is to learn unlabeled samples to discover the intrinsic properties and rules of the data.



Classification

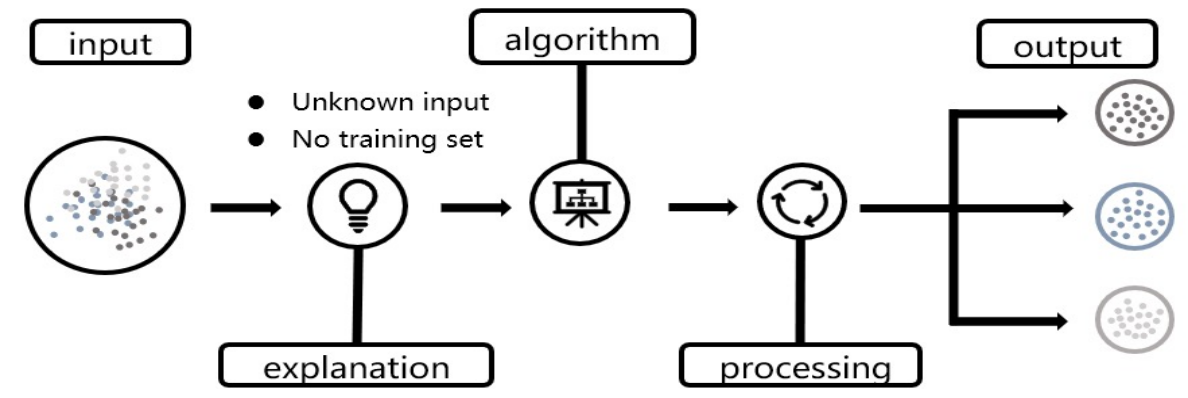


Clustering

Unsupervised Learning

Problems to be solved :

- **Correlation analysis:** finding the probability of different things appearing together.
- **Clustering problem:** dividing similar samples into clusters.
- **Dimensionality reduction analysis:** reduction of data dimensions while retaining meaningful data characteristics, which can improve the accuracy of data modeling and reduce the cost of data storage.



Unsupervised learning process

Summary:The difference between supervised and unsupervised learning

Reference Answer :

1. Supervised learning is from labeled training samples. In supervised learning, we only need to give a set of input samples from which the machine can deduce the possible outcomes of the specified target variable. The machine simply predicts the appropriate model from the input data and calculates the outcome of the target variable from it. The goal to achieve is that "X can predict the variable Y for the input data".

2. Unsupervised learning training learning of unlabeled samples, such as discovering structural knowledge in these samples. In unsupervised learning, target variables such as classification and regression do not exist beforehand. The question to be answered is "what can be found from data X".

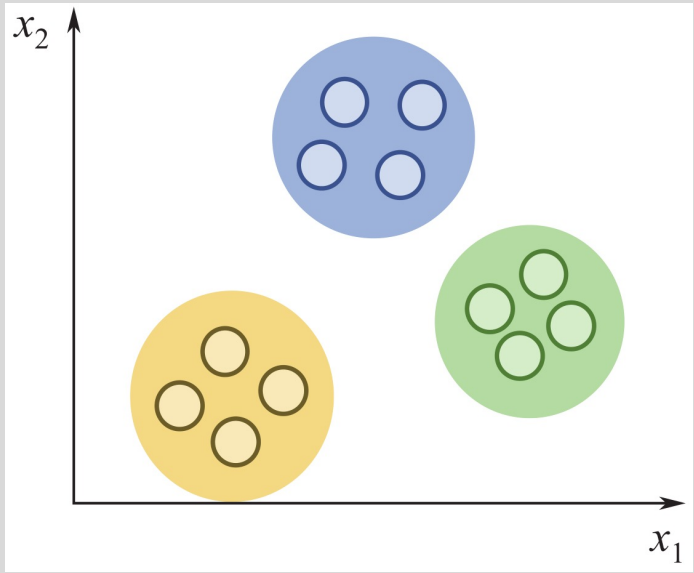
Unsupervised Learning - K-means

K- means

- Simple but most popular partitional algorithm.
- Assume Euclidean space.
- k clusters: $C_1, C_2, \ldots, C_k$
- Minimise the sum of squared distances to the centroid of clusters over all k clusters:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||_2^2$$

which represents the tightness of the samples in the cluster around the cluster mean vector, and the smaller the square error, the higher the similarity of the samples in the cluster.



Clustering

K-means

- **Main Procedure :**

  ① Determine the number of clusters k (plan to divide the data into k classes);

  ② Randomly determine k initial points as the center of mass (randomly selected within the range of data boundaries);

  ③ Calculate the distance to k centroids for each data instance, select the centroid of the smallest distance, and assign it to the cluster corresponding to the centroid until all the data in the data set are allocated to k clusters, and update the centroids of k clusters to the average of all points in the cluster;

  ④ Repeat step ③ to reassign each data instance to a new center of mass until a termination condition is reached, such as no further changes in the results of all data allocations.

K-means

Example of K-means algorithm application

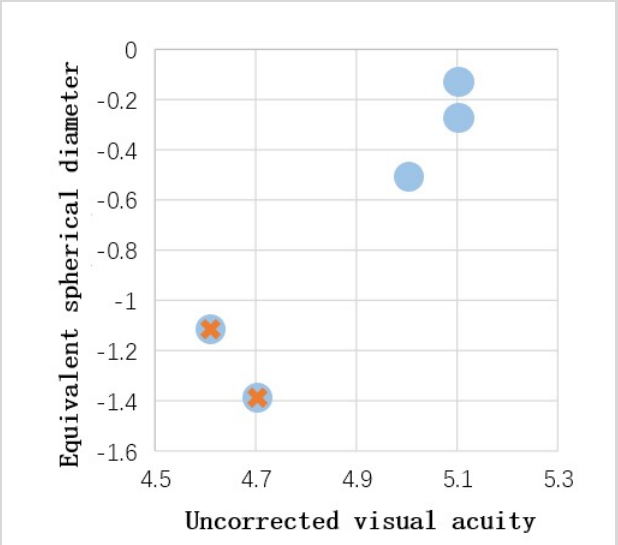
The examination of myopia mainly includes the examination of naked eye vision and equivalent spherical lens. The naked eye visual acuity examination mainly checks the degree of myopia in the naked eye. The equivalent spherical lens is a spherical lens that can convert astigmatism into a similar optical effect, indicating the refractive state of the eye. The following is a cluster analysis of myopia and non-myopia based on naked eye visual acuity and equivalent spherical lens.

Sample number	Uncorrected visual acuity	Equivalent spherical diameter
1	4.7	-1.38
2	4.6	-1.13
3	5	-0.5
4	5.1	-0.25
5	5.1	-0.13

K-means

Example of K-means algorithm application

Step 1: Assume the number of clusters k=2, and randomly select samples 1 and 2 as the initial clustering centers, that is, the initial mean vector $\mu_1 = (4.7, -1.38), \mu_2 = (4.6, -1.13)$. The initial situation is shown in the figure on the right, with the blue circle representing the sample points and the orange cross representing the cluster center.



K-means

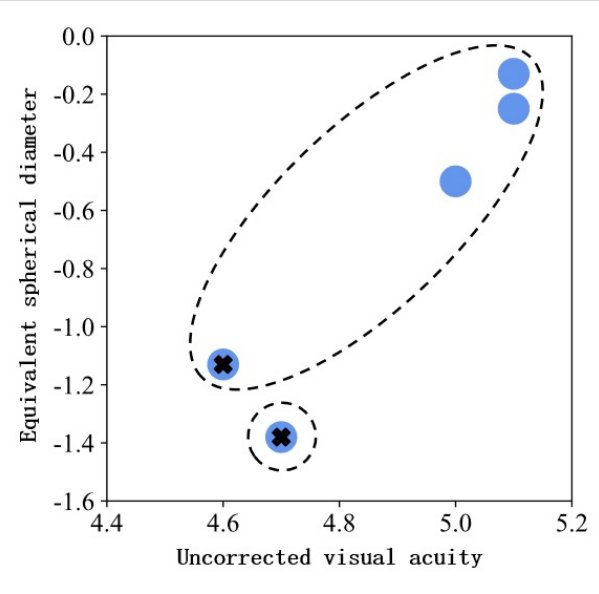
Example of K-means algorithm application

Step 2: Calculate the distance from each sample point to the initial cluster center separately. And regroup according to distance, the results are as follows:

Sample points contained in cluster center A: Sample 1.

Sample points contained in cluster center B: sample 2 ~ sample 5.

Sample number	Cluster center A (4.7, -1.38)	Cluster center B (4.6, -1.13)
1	0	0.2693
2	0.2693	0
3	0.9297	0.7463
4	1.1987	1.0121
5	1.3124	1.1180



K-means

Example of K-means algorithm application

Step 3: according to the result of regrouping computing new clustering center, A clustering center of the mean vector is: $(4.7, -1.38)$, the clustering center B new mean vector $\left(\frac{4.6+5+5.1+5.1}{4}, \frac{-1.13-0.5-0.25-0.13}{3}\right) = (4.95, -0.5025)$. Then recalculate the distance.

Sample number	Cluster center A (4.7, -1.38)	Cluster center B (4.95, -0.5025)
1	0	0.9124
2	0.2693	0.7185
3	0.9297	0.0501
4	1.1987	0.2937
5	1.3124	0.4016

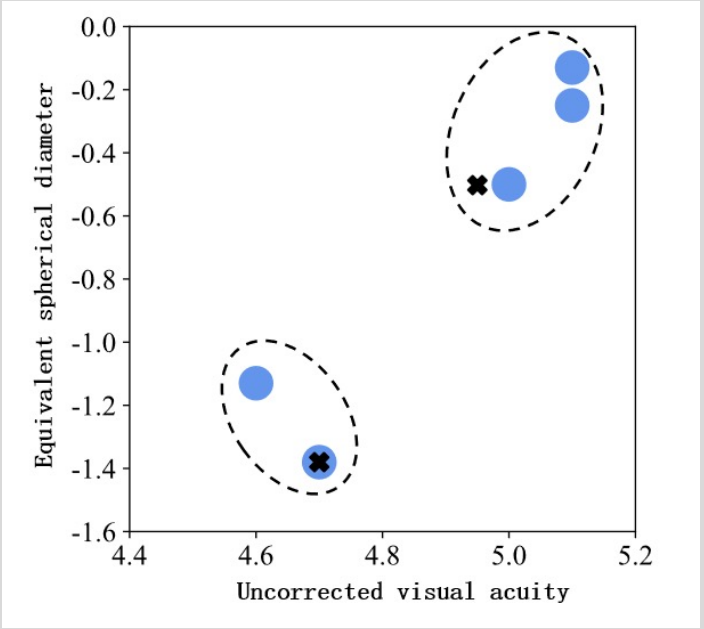
K-means

Example of K-means algorithm application

Step 3: Regroup according to distance, and the result is as follows:

Cluster center A contains sample points: sample 1, sample 2.

Cluster center B contains sample points: sample 3, sample 4, sample 5.



K-means

Example of K-means algorithm application

Step 4: to recalculate the center of mass, the clustering center is A new mean vector is $\left(\frac{4.7+4.6}{2}, \frac{-1.38-1.13}{2}\right) = (4.65, -1.255)$.

New clustering center B mean vector is: $\left(\frac{5+5.1+5.1}{3}, \frac{-0.5-0.25-0.13}{3}\right) = (5.0667, -0.2933)$.

Then recalculate the distance as follows:

Sample number	Cluster center A (4.65, -1.255)	Cluster center B (5.0667, -0.2933)
1	0.1346	1.1469
2	0.1346	0.9581
3	0.8322	0.2172
4	1.1011	0.0546
5	1.2117	0.1667

K-means

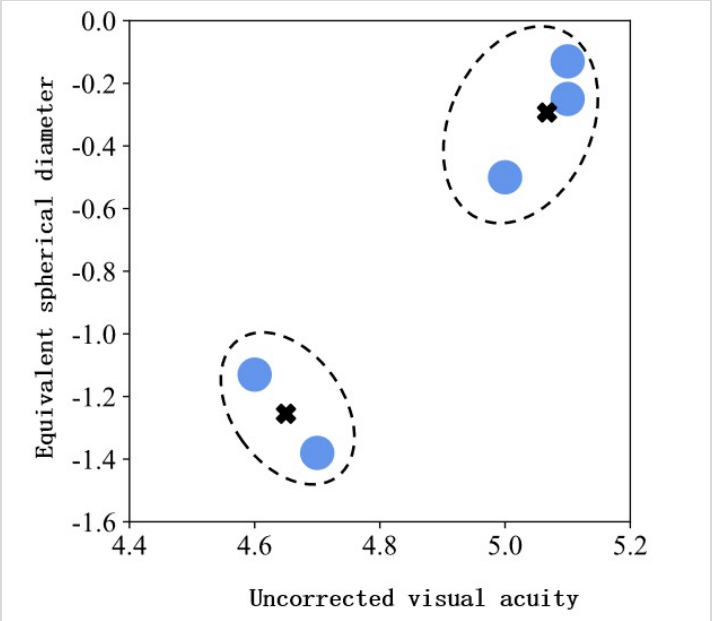
Example of K-means algorithm application

Step 4: Regroup according to distance, the result is as follows:

Cluster center A contains sample points: sample 1, sample 2.

Cluster center B contains sample points: sample 3, sample 4, sample 5.

From then on, the clustering result does not change and the clustering ends. Therefore, the clustering results are shown in the figure on the right. Samples 1 and 2 belong to the same category, and samples 3 to 5 belong to the same category.



Q1: How do K-means deal with empty clusters

In K-means clustering algorithm, empty cluster refers to the case that no sample points are divided into the cluster after a certain round of iteration.

Introduction of AI (CS103)- 09 Machine Learning Algorithms 2

Jimmy Liu 刘江

2023-11-17